

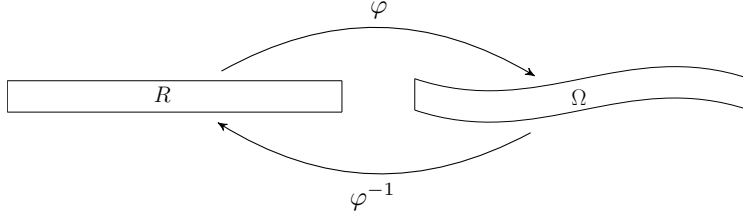
THE STABILITY OF THE FIRST NEUMANN LAPLACIAN EIGENFUNCTION UNDER DOMAIN DEFORMATIONS AND APPLICATIONS

NICHOLAS MARSHALL

ABSTRACT. The robustness of manifold learning methods is often predicated on the stability of the Neumann Laplacian eigenfunctions under deformations of the assumed underlying domain. Indeed, many manifold learning methods are based on approximating the Neumann Laplacian eigenfunctions on an assumed underlying manifold based on data which is viewed through a source of distortion. In this paper, we study the stability of the first Neumann Laplacian eigenfunction in the continuous setting under deformations of the domain by a diffeomorphism. In particular, we are interested in the stability of the first eigenfunction on long thin domains where, intuitively, the first Neumann Laplacian eigenfunction should only depend on the length along the domain. We prove a rigorous version of this statement and apply it to a machine learning problem in geophysical interpretation.

1. INTRODUCTION AND MAIN RESULT

1.1. Introduction. We are motivated by the following empirical observation: on long thin domains, the Neumann Laplacian eigenfunctions demonstrate a high degree of stability with respect to deformations of the domain. In particular, we consider a long thin domain R that has been deformed by some nice diffeomorphism φ resulting in a deformed domain $\Omega = \varphi(R)$.



We say that η_j, v_j and μ_j, u_j are the Neumann (Laplacian) eigenvalues and eigenfunctions on R and Ω , respectively, if they are solutions to the boundary value problems:

$$\begin{cases} -\Delta v_j = \eta_j v_j & \text{in } R \\ \frac{\partial}{\partial n} v_j = 0 & \text{on } \partial R \end{cases} \quad \text{and} \quad \begin{cases} -\Delta u_j = \mu_j u_j & \text{in } \Omega \\ \frac{\partial}{\partial n} u_j = 0 & \text{on } \partial \Omega \end{cases},$$

where Δ denotes the Laplacian, and n denotes an outward normal to the boundary of the domain. Specifically, we are interested in the behavior of the first Neumann eigenfunction on long thin domains, which we illustrate in Figure 1, where the first Neumann eigenfunctions on two different domains have been computed numerically and plotted as color maps.

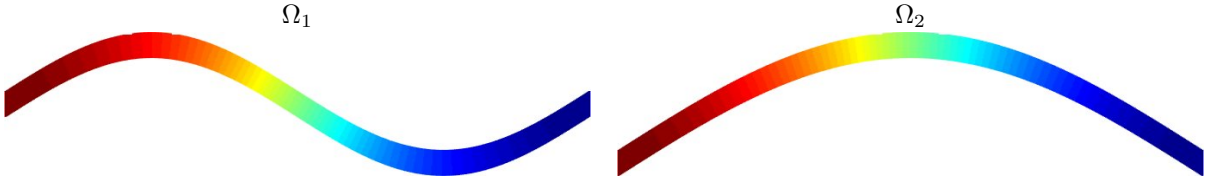


FIGURE 1. The first Neumann Laplacian eigenfunctions on two different long thin domains.

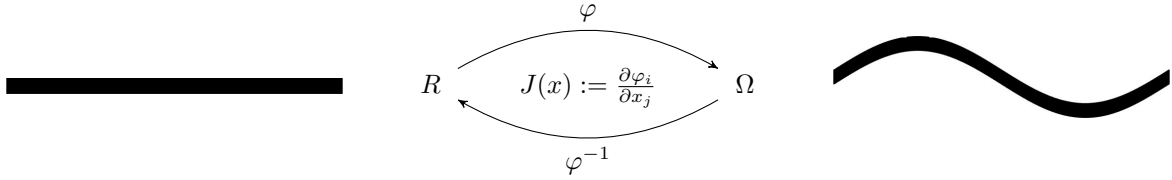
Visually the first Neumann eigenfunctions on Ω_1 and Ω_2 appear highly similar. We quantify the stability of the first Neumann eigenfunction by considering a reference domain R mapped by a diffeomorphism φ to a deformed domain Ω . Intuitively, if the gap between the first the second Neumann Laplacian eigenvalues

on R is large, and the diffeomorphism φ is small (in a sense which will be made precise), then $u_1 \circ \varphi$ (the first Neumann Laplacian eigenfunction on Ω mapped on R) should be roughly equal to v_1 . Alternatively, if the gap between the first and second Neumann Laplacian eigenvalues on R is small, but there is a large gap between the second and third eigenvalues on R , we should still be able to say that $u_1 \circ \varphi$ is roughly some combination of v_1 and v_2 , when φ is small. These observations lead to the following characterization of stability. Let define the quantity

$$E_k := \|(\text{Id} - P_{V_k})u_1 \circ \varphi\|_{L^2(R)} \quad \text{for } k = 1, 2, \dots$$

where $\text{Id} - P_{V_k}$ denotes the projection on the space orthogonal to $V_k = \text{Span}\{v_1, \dots, v_k\}$.

1.2. Main Result. Suppose that R is an open bounded connected subset of \mathbb{R}^d , and assume $|R| = 1$ to fix the scale of the problem. Let $0 = \eta_0 < \eta_1 \leq \eta_2 \leq \dots$, and v_0, v_1, v_2, \dots denote the Neumann Laplacian eigenvalues and eigenfunctions on R . Suppose that $\varphi : R \rightarrow \Omega$ is a diffeomorphism, and suppose $\varphi(x) = y$. Denote by $J(x)$ the Jacobian matrix $\frac{\partial \varphi_i}{\partial x_j}$ of φ , where φ_i denotes the i th component of φ . The determinant of $J(x)$ is denoted by $\det J(x)$, while the minimum and maximum singular values of $J(x)$ are denoted by $\sigma_{\min}(J(x))$ and $\sigma_{\max}(J(x))$, respectively. Let $0 = \mu_0 < \mu_1 \leq \mu_2 \leq \dots$ and u_0, u_1, u_2, \dots , denote the Neumann Laplacian eigenvalues and eigenfunctions on Ω , respectively. We refer to R as the reference domain, i.e., some nice domain where we understand the Neumann eigenfunctions, e.g., a rectangle.



Theorem. Suppose the singular values of the Jacobian matrix $J(x)$ are bounded:

$$\sigma(JJ^\top) \subset [(1 - \varepsilon)^2, (1 + \varepsilon)^2] \quad \text{on } R,$$

where $0 \leq \varepsilon \leq \frac{1}{10}$. Then

$$\|(\text{Id} - P_{V_k})u_1 \circ \varphi\|_{L^2(R)}^2 \leq 20 \cdot \frac{\eta_1 \cdot \varepsilon \cdot d}{\eta_{k+1} - \eta_1} + \varepsilon \cdot d$$

where $\text{Id} - P_{V_k}$ denotes the projection on the space orthogonal to $V_k := \text{Span}\{v_1, v_2, \dots, v_k\}$, $R \subseteq \mathbb{R}^d$ and assuming $\varepsilon \cdot d < 1/10$.

Remark. The following remarks are in order.

- (1) The result of bounding the projection of $u_1 \circ \varphi$ on the space orthogonal to $\bar{V}_k = \text{Span}\{v_0, v_1, \dots, v_k\}$ is the stronger bound

$$\|(Id - P_{\bar{V}_k})u_1 \circ \varphi\|_{L^2(R)}^2 \leq 20 \cdot \frac{\eta_1 \cdot \varepsilon \cdot d}{\eta_{k+1} - \eta_1}.$$

Note that here the projection is taken to the space orthogonal to the space \bar{V}_k , which includes the constant function. As a result, the term $\varepsilon \cdot d$ can be excluded from the bound.

- (2) By symmetry a similar bound on the quantity

$$\|(Id - P_{U_k})v_1 \circ \varphi^{-1}\|_{L^2(\Omega)}$$

exists, where $U_k = \text{Span}\{u_1, \dots, u_k\}$.

- (3) **Example** (Long thin reference domain) Suppose that R is the rectangle $R = [0, 1/\alpha] \times [0, \alpha] \subset \mathbb{R}^2$ for $0 < \alpha < 1$. The Neumann eigenvalues and eigenfunctions on R are of the form

$$(\pi \alpha j)^2 + \left(\frac{\pi \ell}{\alpha}\right)^2, \quad \text{and} \quad 2 \cos(\pi \alpha j x_1) \cos\left(\frac{\pi \ell}{\alpha} x_2\right), \quad \text{for } (x_1, x_2) \in R.$$

Therefore, applying the Theorem to the rectangle in \mathbb{R}^2 yields the following Corollary. Suppose $R = [0, 10] \times [0, 1/10] \subset \mathbb{R}^2$. Then

$$\|P_H u_1 \circ \varphi\|_{L^2(R)}^2 \leq \frac{\varepsilon}{200},$$

where H is the space of function which only depend on the y variable (and are independent of x). The projection on H is bounded by the projection of $u_1 \circ \varphi$ on $Id - P_{\bar{V}_{100}}$, since the first 100 eigenfunction on the rectangle R only vary in the horizontal direction.

1.3. Application. The theoretical results were motivated by a machine learning problem in the field of geophysical interpretation, specifically, problem of organizing the layer structure of seismic images, otherwise known as seismic flattening. Given a seismic image, the objective is to reparameterize the depth function of the pixels in the image such that the layers have constant depth. We will reparameterize the depth function in the image by essentially stretching the seismic image into a thin tall rectangle and computing the first Neumann Laplacian eigenfunction on the stretched domain, see Figure 2 for an example output.

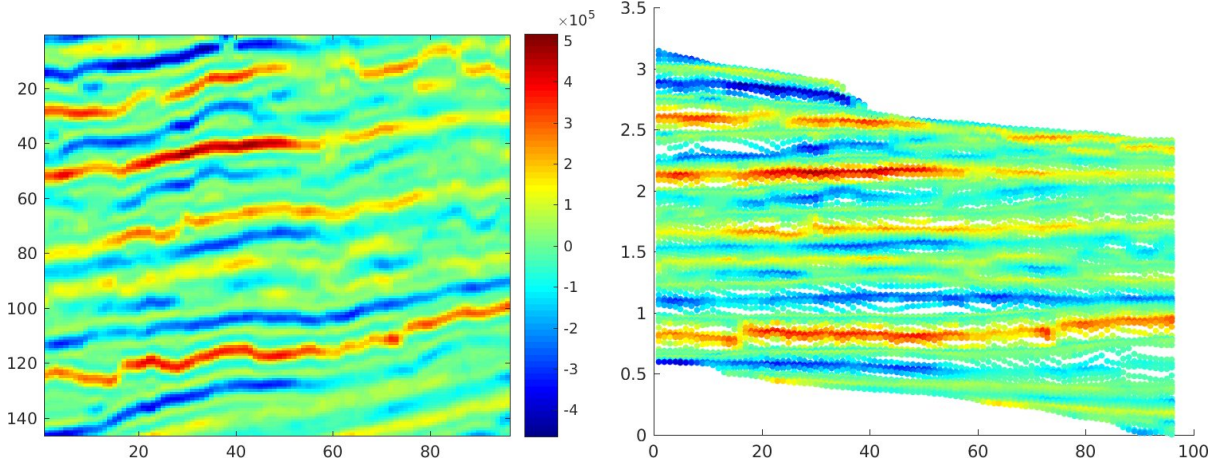


FIGURE 2. Seismic image (left) and result of diffusion flattening (right).

Our approach falls under the class of diffusion geometry machine learning techniques: popular methods of organizing data, which have a theoretical basis in analysis. Diffusion geometry techniques originated with Diffusion Maps [1], and generally construct a diffusion operator on data whose eigenfunctions encode the pertinent geometric information for the given application, see for example [2–7]. In this paper we present a specialized construction of a diffusion operator whose eigenfunctions organize the layers within the seismic image. These eigenfunctions approximate the Neumann Laplacian eigenfunctions on some underlying domain, which we argue is roughly shaped like a tall rectangle. Empirically, we observe that the presented algorithm is very robust. The Theorem proved in this paper acts as a first step to understanding the stability of the algorithm. The proposed algorithm is highly related to other spectral methods such as Laplacian eigenmaps [8], so the Theorem also partially serves to illuminate such techniques.

1.4. Manifold Straightening. To reiterate using different language, the main idea of the application is as follows. Suppose a manifold \mathcal{M} is given whose geometry is described locally by a metric, but no global coordinates are given. Furthermore, suppose the manifold has a smooth vector field, which provides an orientation to each neighborhood. We assume the given manifold is a distorted version of an underlying manifold where the vector field is constant, i.e, all vectors point in the same direction. Our objective is to recover the coordinates for this underlying intrinsic manifold. Our approach has three basic steps:

- (1) Contract the metric in the direction of the vector field.
- (2) Compute the first Neumann Laplacian eigenfunction on the stretched domain.
- (3) Use the first Neumann Laplacian eigenfunction to recover the intrinsic height on the underlying manifold.

In particular, we consider the case where the shape of the underlying domain is known. In this case, knowledge of the shape of the first Neumann eigenfunction on the underlying domain can be used to correctly recover the height function.

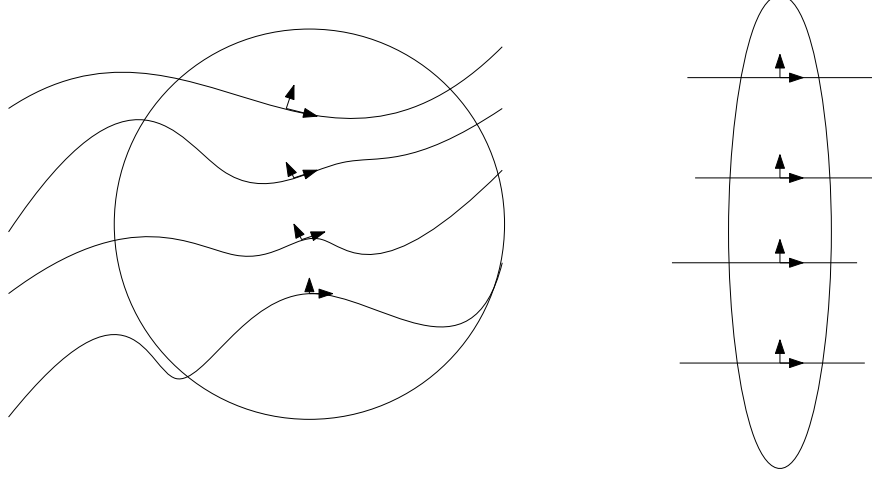


FIGURE 3. Cartoon of a manifold with vector field (left), and result of contracting the metric along the vector field (right).

1.5. Organization. The remainder of the paper is organized as follows. In §2 we introduce notation, outline the main idea of the proof of the Theorem, and provide a detailed proof. In §3 we introduce the application problem, outline the main idea of the algorithm, and display results. In Appendix A, the details of the data adaptive smoothing method, the diffusion operator construction, and the reparameterization method are provided.

2. OUTLINE OF PROOF

2.1. Notation. Suppose R is an open bounded connected subset of \mathbb{R}^d with a piecewise smooth boundary ∂R . We say $0 = \eta_0 < \eta_1 \leq \eta_2 \leq \dots$ and v_0, v_1, v_2, \dots and the Neumann Laplacian eigenvalues and eigenfunctions, respectively, if they are solutions to the following boundary value problem:

$$\begin{cases} \Delta v_j + \eta_j v_j = 0 & \text{in } R \\ \frac{\partial}{\partial n} v_j = 0 & \text{on } \partial R \end{cases},$$

where n denotes an outward normal on the domain R . Suppose that $\varphi : R \rightarrow \Omega$ is a diffeomorphism, and let $y = \varphi(x)$. Denote by J the Jacobian matrix $J(x) = \frac{\partial \varphi_i}{\partial x_j}$ of φ . With this notation, the chain rule for a differentiable function u on Ω can be expressed

$$J^T(\varphi^{-1}(y)) \nabla_y u(y) = \nabla_x u(\varphi(x)),$$

where $\nabla_y = \left(\frac{\partial}{\partial y_1}, \frac{\partial}{\partial y_2} \right)^\top$ denotes the gradient in y , and $\nabla_x = \left(\frac{\partial}{\partial x_1}, \frac{\partial}{\partial x_2} \right)^\top$ denotes the gradient in x . Similarly, for a differentiable function v on R

$$(J^{-1})^T(\varphi^{-1}(x)) \nabla_x v(x) = \nabla_y v(\varphi(y)).$$

Let $\det J(x)$ denote the Jacobian determinant such that

$$\int_{\Omega} u(y) dy = \int_R u(\varphi(x)) |\det J(x)| dx, \quad \text{and} \quad \int_R v(x) dx = \int_{\Omega} v(\varphi^{-1}(y)) |\det J^{-1}(y)| dy$$

express the change of variables for integration. The proof of the Theorem relies on bounding determinant of $J(x)$. The assumed bound on the extreme singular values of $J(x)$ implies that

$$(1 - \varepsilon)^d \leq |\det J(x)| \leq (1 + \varepsilon)^d.$$

Under the assumption $\varepsilon \cdot d < 1/10$ by the mean-value theorem we have the approximation

$$(1 + \varepsilon)^d - 1 \leq (1 + \varepsilon)^{\frac{\varepsilon d}{\varepsilon}} - 1 \leq e^{\varepsilon d} - 1 \leq 2\varepsilon \cdot d.$$

During the course of the proof we will assume, for simplicity of notation, that

$$1 - \delta \leq |\det J(x)| \leq 1 + \delta,$$

and substitute $\delta = 2\varepsilon d$ in the final step to produce the bound. The bound in terms of ε and δ developed in the proof may be useful in cases where an explicit bound on the determinant is given. For example, the Jacobian matrix of a shear transform has arbitrarily large singular values, but the determinant is always 1, e.g., consider the shear transform

$$x = (x_1, x_2) \xrightarrow{\varphi} (x_1 + \alpha x_2, x_2) = (y_1, y_2) = y$$

The Jacobian matrix,

$$J(x) = \begin{pmatrix} 1 & \alpha \\ 0 & 1 \end{pmatrix},$$

has determinant 1, but has arbitrarily large singular values as $\alpha \rightarrow \infty$.

2.2. Outline. Let \gtrsim denote \geq up to multiplication by a constant equal to $1 + O(\varepsilon)$. The proof consists of three main steps. First we will show

$$\eta_1 = \|\nabla_x v_1\|_{L^2(R)}^2 \gtrsim \|\nabla v_1 \circ \varphi^{-1}\|_{L^2(\Omega)}^2 \gtrsim \inf_{\substack{u \perp 1 \\ \|u\|=1}} \|\nabla_y u\|_{L^2(\Omega)}^2 = \mu_1,$$

where the infimum is taken over sufficiently smooth normalized functions which are orthogonal to constant functions on $L^2(\Omega)$. Secondly, we will show that

$$\mu_1 = \|\nabla_y u_1\|_{L^2(\Omega)}^2 \gtrsim \|\nabla_x (u_1 \circ \varphi)\|_{L^2(R)}^2 = \sum_j \eta_j \alpha_j^2,$$

where α_j are the coefficients of $u_1 \circ \varphi$ expanded in the orthogonal basis $\{v_j\}$ of Neumann eigenfunctions on R . Combining these equalities yields

$$\eta_1 \gtrsim \sum_{j=1}^{\infty} \eta_j \alpha_j^2.$$

Intuitively, the sum controls the coefficients α_j^2 for $j = 2, 3, \dots$ with increasing control as η_j increases. A bound on α_0^2 has to be developed separately. In the following we will make this argument precise and compute the constants associated with the \gtrsim inequalities.

2.3. Proof of Theorem.

Proof. First we establish an upper bound on μ_1 in terms of η_1 . Multiplying the equation $\Delta v_1 + \eta_1 v_1 = 0$ by v_1 , integrating over R , and applying Green's Identity yields

$$\eta_1 = \|\nabla_x v_1\|_{L^2(R)}^2,$$

which by the chain rule, and a change of variables of integration is equivalent to

$$\eta_1 = \|J^T \circ \varphi^{-1} \nabla_y \circ \varphi^{-1} \sqrt{|\det J^{-1}|}\|_{L^2(\Omega)}^2.$$

Therefore by the assumed bounds for $\sigma_{\max}(J)$ and $|\det J|$, we conclude that

$$\eta_1 \cdot \frac{(1 + \varepsilon)^2}{1 - \delta} \geq \|\nabla_y v_1 \circ \varphi^{-1}\|_{L^2(\Omega)}^2.$$

By the minimax principle

$$\mu_1 = \inf_{u \perp 1} \frac{\|\nabla_y u\|_{L^2(\Omega)}^2}{\|u\|_{L^2(\Omega)}^2},$$

where the infimum is taken over differentiable functions u , which are orthogonal to constant functions on $L^2(\Omega)$. Therefore

$$\frac{\|\nabla(v_1 \circ \varphi^{-1} - c)\|_{L^2(\Omega)}^2}{\|v_1 \circ \varphi^{-1} - c\|_{L^2(\Omega)}^2} \geq \mu_1,$$

where c is a constant such that the function $v_1 \circ \varphi^{-1} - c$ is orthogonal to constant functions on $L^2(\Omega)$, i.e.,

$$c = \frac{1}{|\Omega|} \int_{\Omega} v_1 \circ \varphi^{-1} dx.$$

For now, combining this inequality with the previous inequality comparing η_1 and $\|\nabla_y v_1 \circ \varphi^{-1}\|_{L^2(\Omega)}^2$ yields

$$\eta_1 \frac{(1+\epsilon)^2}{(1-\delta)} \frac{1}{\|v_1 \circ \varphi^{-1} - c\|_{L^2(\Omega)}^2} \geq \frac{\|\nabla_y(v_1 \circ \varphi^{-1} - c)\|_{L^2(\Omega)}^2}{\|v_1 \circ \varphi^{-1} - c\|_{L^2(\Omega)}^2} \geq \mu_1.$$

To complete this inequality, it remains to compute a lower bound on $\|v_1 \circ \varphi^{-1} - c\|_{L^2(\Omega)}^2$. The constant c can be computed by taking the inner product of $v_1 \circ \varphi^{-1}$ with the constant function $|\Omega|^{-1}$:

$$c = |\Omega|^{-1} \int_{\Omega} v_1(\varphi^{-1}(y)) dy = |\Omega|^{-1} \int_R v_1(x) |\det J(x)| dx \leq \delta |\Omega|^{-1} \int_R |v_1(x)| dx \leq \delta |\Omega|^{-1} |R| \leq \frac{\delta}{1-\delta}.$$

Since $v_1 \circ \varphi^{-1} - c$ is orthogonal to constant functions in $L^2(\Omega)$,

$$\begin{aligned} \|v_1 \circ \varphi^{-1} - c\|_{L^2(\Omega)}^2 &= \|v_1 \circ \varphi^{-1}\|_{L^2(\Omega)}^2 - \|c\|_{L^2(\Omega)}^2 \geq \int_R |v_1(x)|^2 |\det J(x)| dx - |\Omega| \left(\frac{\delta}{1-\delta} \right)^2 \\ &\geq (1-\delta) - (1+\delta) \frac{\delta^2}{(1-\delta)^2} = \frac{(1-\delta)^3 - (1+\delta)\delta^2}{(1-\delta)^2}. \end{aligned}$$

Substituting this lower bound into our previous inequality yields:

$$\eta_1 \frac{(1+\epsilon)^2(1-\delta)}{(1-\delta)^3 - (1+\delta)\delta^2} \geq \mu_1.$$

Next we express μ_1 in terms of an integral of the function $\mu_1 \circ \varphi$ over R ,

$$\mu_1 = \|(J^{-1})^T \circ \varphi \nabla_x(u \circ \varphi) \sqrt{|\det J|}\|_{L^2(R)}^2.$$

This construction is symmetric to our equation for η_1 in terms of an integral of $v_1 \circ \varphi^{-1}$ over Ω . Applying the bounds on $\sigma_{\min}(J)$ and $|\det J|$ yields

$$\mu_1 \frac{(1+\delta)}{(1-\epsilon)^2} \geq \|\nabla_x(u_1 \circ \varphi)\|_{L^2(R)}^2.$$

Since the Neumann Laplacian eigenfunctions $\{v_j\}$ form an orthonormal basis of $L^2(R)$, we can expand

$$u_1 \circ \varphi(x) = \sum_{j=0}^{\infty} \alpha_j v_j(x),$$

where

$$\alpha_j = \int_R u_1(\varphi(x)) v_j(x) dx.$$

By Green's first identity, and the orthogonality of the eigenfunctions $\{v_j\}$

$$\left\| \nabla_x \sum_j \alpha_j v_j \right\|_{L^2(R)}^2 = \sum_j \eta_j \alpha_j^2.$$

Combining the previously develop inequalities and expanding $u_1 \circ \varphi$ in $\{v_j\}$ establishes the inequality

$$\eta_1 E_{\delta,\epsilon} \geq \sum_{j=1}^{\infty} \eta_j \alpha_j^2.$$

where

$$E_{\delta,\epsilon} = \frac{(1+\epsilon)^2}{(1-\epsilon)^2} \frac{(1+\delta)(1-\delta)}{(1-\delta)^3 - (1+\delta)\delta^2}.$$

Notice that the sum excludes the term $\eta_0 \alpha_0^2$ since $\eta_0 = 0$. Therefore, our inequality offers no control over α_0^2 . However, since u_1 is orthogonal to constants, we suspect $u_1 \circ \varphi$ is nearly so on $L^2(R)$. Before proceeding we will develop specific bounds for α_0^2 . By definition,

$$\alpha_0^2 = \left(\int_R u_1(\varphi(x)) dx \right)^2 = \left(\int_\Omega u_1(y) |\det J^{-1}(y)| dy \right)^2 \leq \left(\frac{\delta}{1-\delta} |\Omega| \right)^2 \leq \frac{\delta^2(1+\delta)^2}{(1-\delta)^2}.$$

Up to now, we have developed the necessary pieces to prove the Theorem. We begin with the inequality

$$\eta_1 E_{\delta, \varepsilon} \geq \sum_{j=1}^{\infty} \eta_j \alpha_j^2,$$

and subtract $\sum_{j=1}^k \eta_j \alpha_j^2$ from each side yielding

$$\eta_1 E_{\varepsilon, \delta} - \sum_{j=1}^k \eta_j \alpha_j^2 \geq \sum_{j=k+1}^{\infty} \eta_j \alpha_j^2.$$

Then since the η_j are in ascending order

$$\eta_1 \left(E_{\varepsilon, \delta} - \sum_{j=1}^k \alpha_j^2 \right) \geq \eta_{k+1} \sum_{j=k+1}^{\infty} \alpha_j^2.$$

Since $\frac{1}{1+\delta} \leq \|u_1 \circ \varphi\|_{L^2(R)} = \sum_{j=0}^{\infty} \alpha_j^2$,

$$\eta_1 \left(E_{\varepsilon, \delta} - \left(\frac{1}{1+\delta} - \sum_{j=k}^{\infty} \alpha_j^2 - \alpha_0 \right) \right) \geq \eta_k \sum_{j=k}^{\infty} \alpha_j^2.$$

Rearranging terms yields

$$\frac{\eta_1 F_{\varepsilon, \delta}}{\eta_k - \eta_1} \geq \sum_{j=k}^{\infty} \alpha_j^2$$

where

$$F_{\varepsilon, \delta} = \frac{(1+\varepsilon)^2}{(1-\varepsilon)^2} \cdot \frac{(1+\delta)(1-\delta)}{(1-\delta)^3 + (1+\delta)\delta^2} + \frac{\delta^2(1+\delta)^2}{(1-\delta)^2} - \frac{1}{1+\delta}.$$

Now suppose that $\delta = 2\varepsilon d$. Expanding $F_{\delta, \varepsilon}$ is a power series and assuming $\varepsilon d < 1/10$ gives,

$$20 \cdot \frac{\eta_1 \cdot \varepsilon \cdot d}{\eta_k - \eta_1} + \varepsilon d \geq \alpha_0^2 + \sum_{j=k}^{\infty} \alpha_j^2$$

as was to be shown. □

3. APPLICATIONS

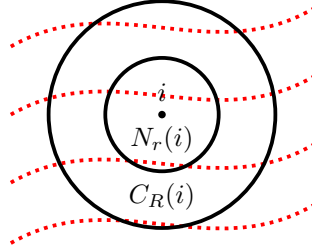
3.1. Problem. In the field of geophysical interpretation the problem of organizing the layers of a seismic image, otherwise known as seismic flattening, is of interest [9–11]. Specifically, given a seismic image, the objective is to modify the depth function of the seismic image such that each layer has constant depth. Since each layer was deposited at roughly the same time, the depth function which flattens the layers is sometimes referred to as geological time. In general, a seismic image consists of an $m \times n \times l$ real-valued tensor whose first coordinate is referred to as depth. As the depth increases, the values in the tensor oscillate between positive and negative values, but in a nonuniform way. For example in Figure 2 a two dimensional slice of a three dimensional seismic image is plotted using a color map, which exhibits this oscillating behavior. In the following, we will primarily restrict our attention to such two dimensional slices of seismic tensors, except for the filtering process on the data, which does incorporate the pixels surrounding our two dimensional slice in the tensor.

3.2. Approach Outline. We approach the problem of seismic flattening from the diffusion geometry perspective. That is to say, we construct a diffusion process on the seismic image whose eigenfunctions emphasize the geometric features of interest, which, in this case, are the layers of the image. Our method has three main steps:

- (1) **Adaptive Filtering** A Principal Component Analysis filtering technique is used to associated filtered feature vectors to each pixel in the two dimensional seismic image. These filtered features incorporate information from the surrounding pixels.
- (2) **Kernel Construction** A diffusion kernel is defined, which propagates rapidly along the layers of the seismic image and slowly perpendicular to these layers. This kernel is defined using the filtered feature vectors.
- (3) **Layer Organization** The first eigenfunction of the diffusion operator (which should resemble the first Neumann Laplacian eigenfunction on the intrinsic underlying domain that is roughly shaped like a tall thin rectangle) is then used to organized the layers of the image.

Intuitively, since the constructed diffusion process propagates rapidly along the layers of the seismic image, and slowly perpendicular to the layers, if the diffusion operator is taken to a sufficient power, diffusion starting at a single point on a layer will completely propagate along the layer, with very little propagation perpendicular to the layer. This powered diffusion operator essentially replaces functions with their average along the layers of the seismic image. However, the eigenfunctions of the original diffusion operator and the powered diffusion operator are the same. Therefore, the first nontrivial eigenfunction of the diffusion operator must be essentially equal to its average on the layers of the image. That is to say, the first eigenfunction must be essentially constant on the layers of the image. A similar idea was used by Lafon [1]. A complete description of the proposed method is provided in Appendix A. Now we will just focus on explaining the kernel construction, which is the main idea of the algorithm.

3.3. Main idea. The main idea of the algorithm consists of the kernel construction. Suppose filtered feature vectors $f(i)$, which summarize the structure of the seismic image at i , have already been computed using the method described in Appendix A. For each pixel i we consider two spatial neighborhoods: a calibration neighborhood C_R , and a propagation neighborhood N_r , where $0 < r < R$ denotes the radius of each neighborhood.



The calibration neighborhood $C_R(i)$ determines the level of variation of the filtered feature vectors around pixel i . Specifically, we define:

$$M(i) = \max_{j \in C_R(i)} \|f(i) - f(j)\|_2^2.$$

This function $M(i)$ accounts for the differing levels of variation across the seismic image. The propagation neighborhood $N_r(p)$ determines the support of the diffusion operator. Specifically, we define the kernel

$$W(i, j) = \exp\left(-\frac{\|f(i) - f(j)\|_2^2}{\varepsilon M(p)}\right) \quad \text{for } j \in N_r(i)$$

and $W(i, j) = 0$ if $j \notin N_r(i)$, where $\varepsilon > 0$ is a parameter. Restricting the propagation of the diffusion process to the small neighborhood N_r ensures the resulting kernel will be sparse, and hence computationally efficient to work with. By appropriately choosing the parameter $\varepsilon > 0$, we can control how much the truncation to $N_r(i)$ effects the diffusion operator. The filtered features are designed in such a way that they are relatively constant when a pixel is translated along a layer, and differ greater as a pixel moves perpendicular to a layer (The pixel values themselves also have this property, but with more noise). Therefore, by choosing

the parameters, R , r , and ε are appropriately, the diffusion process will strongly prefer to travel along the level lines of the image, and be strongly penalized for traveling perpendicular to these level lines. For this constructed diffusion kernel, the underlying domain on which the diffusion kernel corresponds to isotropic diffusion must be very narrow and tall, hence the connection to the Theorem proved earlier in the paper. The details of this construction are included in Appendix A.

3.4. Example. In this section we will illustrate each step of the described method.

Filtered Features. First, each pixel of the selected two dimensional slice is associated with a filtered feature vector $f(i)$ which incorporates information from the pixels surrounding i in the seismic tensor. The filtered feature vector summarize the local structure of the seismic image in a smooth way. In Figure 4 the two dimensional slice of the seismic image is plotted verses one of the coordinates of the filtered feature vector $f(i)$. This data adaptive filtering process serves to remove local noise and incorporate some of the surrounding structures.

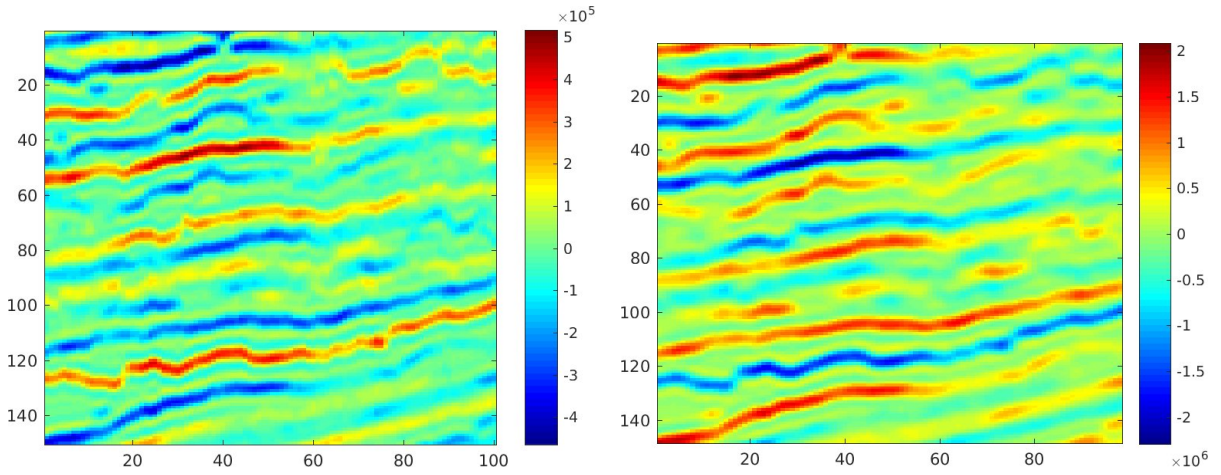


FIGURE 4. Seismic image (left), and coordinate of filtered feature vector (right).

Diffusion operator eigenfunctions. Next, we define the kernel $W(i, j)$ using the filtered features vectors and normalized $W(i, j)$ appropriately to define a diffusion operator $P(i, j)$. The first two (nontrivial) eigenvectors of P are plotted in Figure 5. As expected the first eigenvector is approximately monotone, while the second eigenvector resembles a higher order oscillation. The subsequent eigenvectors additionally contain oscillations in the horizontal direction. Therefore, the underlying domain corresponds to some tall rectangular domain whose height is approximate 2 – 3 times its width.

Flattening. Given the first eigenfunction of the diffusion operator P , we can recover the depth function by assuming this eigenfunction coincides with the first Neumann Laplacian eigenfunctions on some tall thin rectangle. The result of this flattening procedure is plotted in Figure 6.

Magnified flattening. Comparing the scatter plot to the original seismic image in Figure 6 its evident that the layers are already fairly flat. However, at this resolution, the markers in the scatter plot are overlapping. By zooming into the scatter plot and the corresponding area in the seismic image, we can see that the description generated by ψ_1 is in fact far richer than simply flattening. In Figure 7 we zoom into the y -axis of Figure 6 from 90 to 130. At this new level of zoom white space appears between the scatter plot markers. This white space is simply a result of the markers diminishing in size, and demonstrates that the diffusion flattening procedure offers an advantageous side effect: the points within a layer are automatically clustered by the diffusion process.

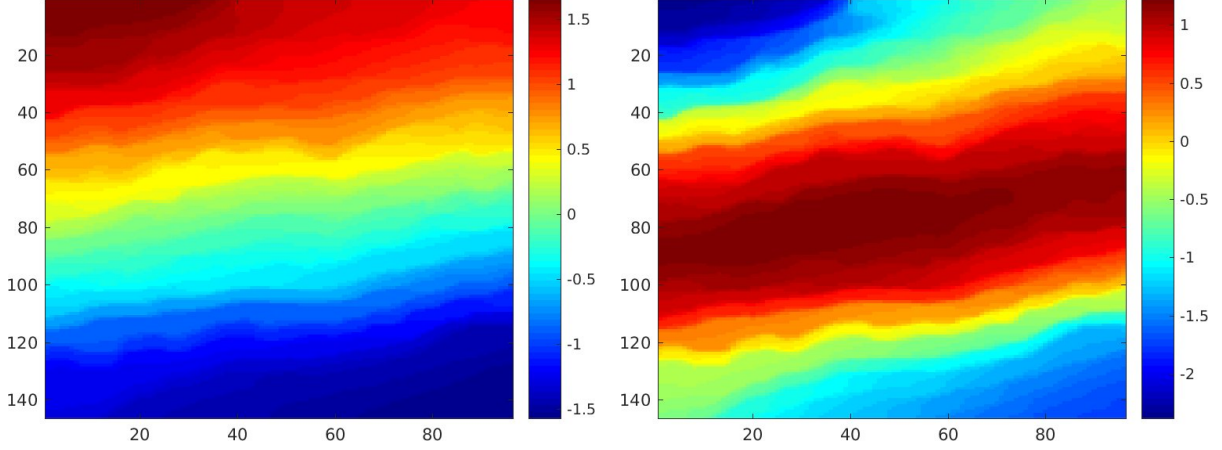


FIGURE 5. First (left) and second (right) eigenfunction of the diffusion operator P .

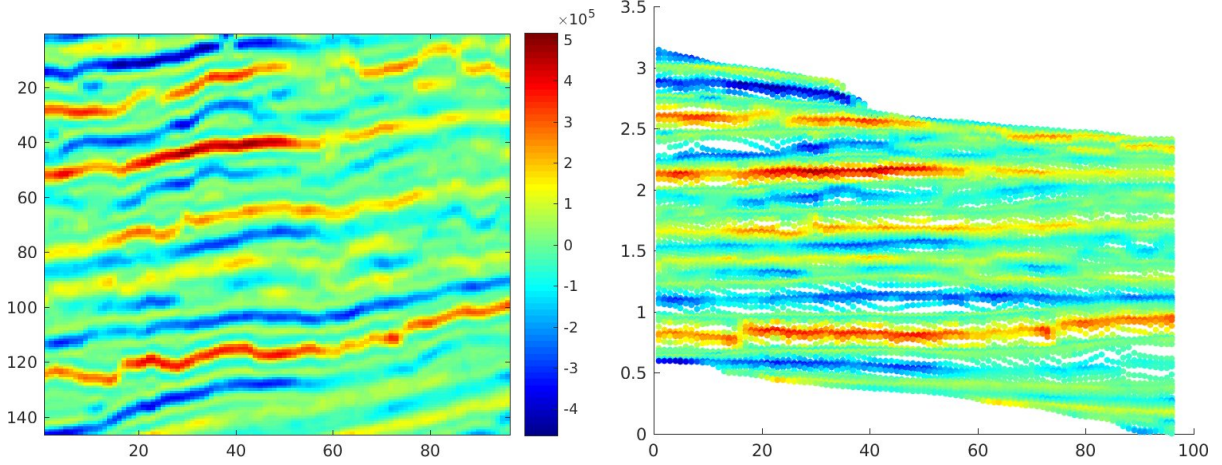


FIGURE 6. Seismic image (left), and result of diffusion flattening (right).

Summary. In summary, given a two dimensional seismic image we have defined a diffusion process on the image using structural information from the surrounding pixels. The diffusion operator was defined such that the resulting diffusion process propagates rapidly along the layers of the seismic image and slowly perpendicular to these layers. Therefore, the diffusion should corresponds to isotropic diffusion on some underlying domain, roughly shaped like a tall thin rectangle where the layers of the seismic image are organized. Therefore, using our intuition backed by the Theorem proved earlier in the paper, we can recover the depth function in the image by assuming the first eigenfunction of the diffusion operator resembles the first Neumann Laplacian eigenfunction on the rectangle.

APPENDIX A. ALGORITHM DETAILS

A.1. Adaptive Filtering. In this section we describe the adaptive filtering method. The filtering method is described for a general three dimension image since it requires no additional effort, whereas for the rest of the algorithm we are mainly interested in the filtered feature vectors computed for a two dimensional slice of a three dimensional seismic image. Let X denote a three dimension seismic image, i denote a pixel from X , and $v(i)$ denote the value associated with pixel i . Each interior pixel i of the seismic image is surrounded by a $3 \times 3 \times 3$ cube of pixels in X . Given an interior pixel i , let $g(i)$ denote the set of values associated with the pixels surrounding i . By fixing a method of rearranging a $3 \times 3 \times 3$ cube of pixels into a vector, we can consider each $g(i)$ as a vector in \mathbb{R}^{27} . Given a collection of feature vectors $g(i)$, we apply principal

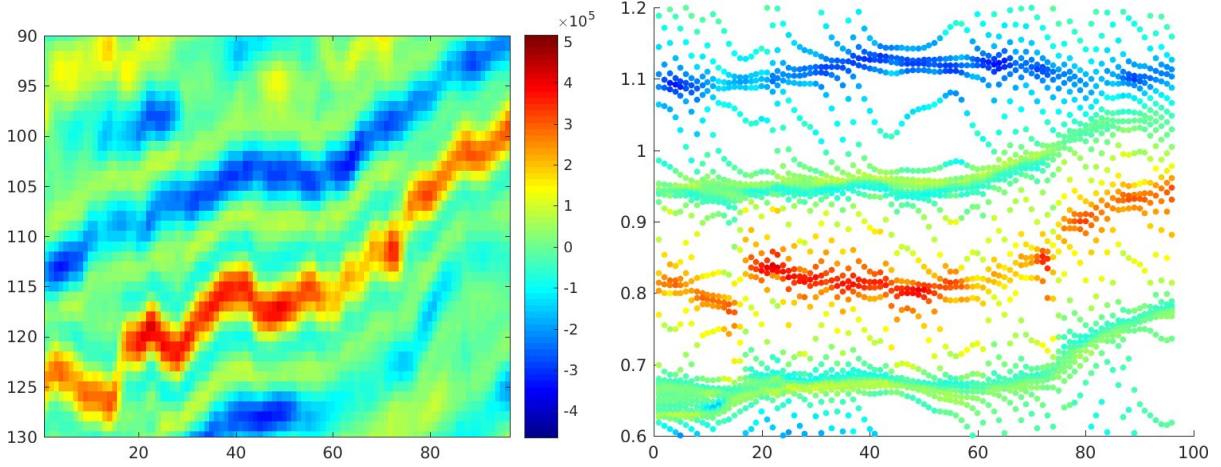


FIGURE 7. Seismic image Y with y -axis zoomed into 90 to 130, and corresponding region of flattened image. At this zoom level, the detailed structure of the diffusion layer organization becomes evident.

component analysis (PCA) to create a low dimension description of each $g(i)$. Specifically, let A denote the $27 \times N$ matrix whose columns consist of the feature vectors $g(i)$ for each of the N interior pixels. Define the feature covariance matrix

$$C = \frac{1}{1 - N} \left(A - A \frac{1}{N} \mathbf{1} \mathbf{1}^T \right) \left(A - A \frac{1}{N} \mathbf{1} \mathbf{1}^T \right)^T,$$

where $\mathbf{1}$ denotes a column vector of all ones in \mathbb{R}^N . Let u_1 denote the principal component of C , that is, the eigenvector of C associated with the largest eigenvalue. The coordinates $w(i)$ of $g(i)$ in the principal component u_1 are defined

$$w(i) = u_1^T g(i)$$

for each pixel i . We refer to $w(i)$ as the filtered coordinates of X . Let $f(i)$ denote the vector in \mathbb{R}^{27} corresponding to the filtered pixel values of the pixels in the $3 \times 3 \times 3$ cube of pixels surrounding pixel i . As before, the $3 \times 3 \times 3$ cubes of pixels are converted into vectors via some fixed method of rearrangement. The filtered features $f(i)$ are used to define the diffusion process in the following section.

Summary. In summary, to each pixel i we associated the cube of surrounding pixel values $g(i)$, and used PCA to summarize each cube by a single coordinate $w(i)$. Finally, we associated each pixel with a filtered feature vector $f(i)$, corresponding to a cube of the surrounding filtered coordinates $w(i)$.

A.2. Kernel Construction. In this section we describe the construction of a diffusion process on the pixels of a seismic image, which propagates rapidly along the layers of the image and slowly perpendicular to these layers. For simplicity we restrict our attention to a two dimension slice of a three dimensional seismic image, as shown in the left side of Figure 4.

Let Y denote a two dimensional slice of a given three dimensional seismic image, i denote a pixel of Y , and $f(i)$ denote the filtered feature vector $f(i) \in \mathbb{R}^{27}$ whose construction is described in the previous section. Let $x(i)$ denote the spatial coordinates of pixel i in Y . For example, if Y is of dimension $m \times n$, then $x(i) = (x_1(i), x_2(i)) \in \{1, \dots, m\} \times \{1, \dots, n\}$. For each pixel $i \in Y$, let $N_r(i)$ denote the collection of pixels

$$N_r(i) = \{j : \|x(i) - x(j)\|_2 < r\},$$

where $r > 0$ is some small radius, e.g., $r = 2$. We will refer to $N_r(i)$ as the propagation neighborhood. In addition to $N_r(i)$ we define a larger calibration neighborhood $C_R(i)$ for $R > r$, which is coarsely sampled. Specifically, if $(x_1(i), x_2(i)) = x(i)$ denote the spatial coordinates of pixel i , then define

$$C_R(i) = \{j : \|x(j) - x(i)\|_2 < R, x_1(j) - x_1(i) \equiv 0 \pmod{3}, x_2(j) - x_2(i) \equiv 0 \pmod{3}\},$$

so that $C_R(i)$ consists of the pixels whose $3 \times 3 \times 3$ feature cubes are non-overlapping, and whose spatial distance is less than R from pixel i . Using the calibration neighborhood $C_R(i)$ we define

$$M(i) = \max_{j \in C_R(i)} \|f(i) - f(j)\|_2^2,$$

and define the asymmetric kernel $W_\varepsilon(i, j)$ by

$$W_\varepsilon(i, j) = \exp\left(-\frac{\|f(i) - f(j)\|_2^2}{\varepsilon \cdot M(i)}\right) \quad \text{if } j \in N_r(i),$$

and

$$W_\varepsilon(i, j) = 0 \quad \text{if } j \notin N_r(i).$$

The purpose of defining the calibration neighborhood $C_R(i)$ rather than just normalizing the kernel over $N_r(i)$ is to avoid the case where the entire neighborhood $N_r(i)$ is contained in a single layer. In this case, all pixels within $N_r(i)$ should be strongly connected, but normalizing using $N_r(i)$ would emphasize minute differences in the feature vectors. Of course, this problem could be avoided by increasing the radius r ; however, increasing r increases the number of nonzero entries in each row of the kernel W by a factor of r^2 : better to define a separate calibration neighborhood $C_R(i)$ to avoid both the normalization and complexity issues. As defined, kernel $W_\varepsilon(i, j)$ takes a maximum value of 1 when $f(i) = f(j)$, and a minimum value of

$$W_\varepsilon(i, j) = \exp\left(-\frac{1}{\varepsilon}\right)$$

if $\|f(i) - f(j)\|_2^2 = M(i)$. Therefore, rather than choosing $\varepsilon > 0$ we parameterize our model by $\delta > 0$ and define

$$\varepsilon = -\frac{1}{\log \delta}$$

With this parameterization, the minimum possible value of the kernel in each neighborhood is δ . In order to be compatible with the standard Diffusion Maps framework described in [12], the kernel used to define the diffusion process must be symmetric. Therefore, we define a symmetrized version $K(i, j)$ of $W_\varepsilon(i, j)$ by

$$K(i, j) = \frac{1}{2} (W_\varepsilon(i, j) + W_\varepsilon(j, i))$$

Notice that the dependence on ε of K (which in turn depends on δ) was suppressed for notational simplicity.

Summary. We first choose a radius $r > 0$ which defines a spatially local propagation neighborhood. Second, we choose a radius $R > 0$ which defines a coarse calibration neighborhood. Third, we choose a parameter $\delta > 0$, which determines the minimum possible affinity each in neighborhood. Finally, we symmetrize the constructed kernel, resulting in the symmetric kernel $K(i, j)$. In the following section, we use the kernel K in conjunction with the standard Diffusion Maps framework [12] to create a description of the layer geometry of the image Y .

A.3. Layer Organization. In this section we describe how the kernel K defined in the previous section can be used to generate a description of the layer structure of Y . In particular, we use the ($\alpha = 0$) diffusion kernel construction of Diffusion Maps [12] in conjunction with a randomized matrix decomposition technique as introduced by [13], as well as intuition about the form of the Laplacian-Neumann eigenvectors, to define a new depth function h of the pixels of the seismic image Y .

Given the kernel $K(i, j)$, we define the diffusion operator P by

$$P(i, j) = \frac{K(i, j)}{q(i)},$$

where

$$q(i) = \sum_j K(i, j).$$

The matrix operator $P(i, j)$ is row stochastic; therefore, P is a Markov operator when applied from the right, and a diffusion or averaging operator when applied from the left. Since P is similar to the symmetric matrix A ,

$$A = Q^{1/2} P Q^{-1/2},$$

where

$$Q = \text{diag}(q),$$

the matrix P is diagonalizable. Moreover, the eigenvectors ψ_j of P are exactly given by

$$\psi_j = Q^{-1/2} \phi_j$$

where ϕ_j is an eigenvector of A of the same eigenvalue λ_j .

Therefore, in order to compute the eigenvectors of P , it suffices to decompose A and apply $Q^{-1/2}$. From a computational point of view, working with A is preferable since A is symmetric and has operator norm 1. Furthermore, the operator A is sparse by construction with most $2 \cdot |N_r(i)|$ nonzero entries in each row i , and is of finite rank invariant to the sampling density of the data (at least when the kernel bandwidth is fixed). Therefore, randomized matrix decomposition algorithms, such as described in [14], provide an efficient method to compute the top few eigenvectors of A . Given these eigenvectors, the eigenvectors of P can be recovered by applying $Q^{-1/2}$ as previously mentioned.

By construction, the eigenvectors of P approximate the Neumann Laplacian eigenfunctions of the underlying geometry of the pixels of Y on which P corresponds to isotropic diffusion, cf. [12]. The largest eigenvalue λ_0 of P is 1, which corresponds to a constant eigenvector $\psi_0 = 1$. However, the second eigenvector ψ_1 is of interest. Recall that P is based on the kernel K , which is defined to be large (i.e., close to 1) between similar pixels within a spatial neighborhood, and on the order of a small constant δ (e.g., $\delta = 10^{-7}$) between highly dissimilar pixels with respect to the calibration neighborhood. While moving along a layer in the seismic image Y , the pixels (as characterized by the feature vector $f(i)$) should be highly similar. However, when moving perpendicular to a layer, the feature vectors $f(i)$ will change rapidly, and pixels quickly become highly dissimilar to the starting point. As a result, the diffusion described by P travels rapidly along the layers of Y , and slowly perpendicular to these layers. That is to say, the underlying geometry of the pixels on which P represents isotropic diffusion must have a relatively small distance between pixels on the same layer when compared to pixels on different layers, which are equally spatially distance in Y .

Therefore, the underlying geometry of P roughly corresponds to a tall rectangle shape, that is, a rectangle whose height is greater than its width. Of course, due to the variation in the earth, the underlying domain will not be exactly rectangle shaped, but resemble some type of deformed tall rectangle where the layers run horizontally and are relatively flat. In Section 2, we proved an explicit stability result, which indicates that the first Neumann Laplacian eigenfunctions on tall rectangle are highly stable under deformation on the underlying domain. This proof is consistent with numerical experiments. Intuitively, the stability results from the fact that if a rectangle's height is at least twice its width, the first couple Neumann Laplacian eigenfunctions will only oscillate vertically, which provides a buffer to any horizontal oscillations.

Recall, that the first Neumann Laplacian eigenfunction on the rectangle $[0, \varepsilon] \times [0, 1]$ where $1 > \varepsilon > 0$ is

$$\psi_1(x, y) = \cos(\pi y)$$

Assuming that ψ_1 is of a similar form, the height function on the underlying domain can be recovered by

$$h(i) = \arccos \left(\frac{\psi_1(i) - \min_j \psi_1(j)}{\max_j \psi_1(j) - \min_j \psi_1(j)} \right)$$

up to linear scaling. Depending on the height of the underlying rectangular domain, the next couple eigenfunctions ψ_2, ψ_3 , etc. should resemble $\cos \pi 2y, \cos \pi 3y$, etc., until the first eigenvector emerges with horizontal oscillations, after which the eigenfunctions likely become difficult to interpret because of noise.

Summary. We have constructed a diffusion operator P whose eigenfunctions approximate the Neumann Laplacian eigenfunctions on the underlying domain, which by the construction of the kernel K should resemble a tall rectangle. Using randomized matrix decomposition techniques, the eigenfunctions of P can be computed, and the arccos function can be used to recover the height function of the underlying rectangular domain. This height function should organize the layers of the image.

Acknowledgements. We would like to thank Anthony Vassiliou and Lionel Woog for the seismic data and for bringing the application problem to our attention. Furthermore, we would like to thank Ronald Coifman for numerous insightful conversations, and Stefan Steinerberger for many helpful comments and suggestions.

REFERENCES

- [1] S. Lafon, *Diffusion Maps and Geometric Harmonics*. PhD thesis, Yale University, 2004.
- [2] R. R. Lederman and R. Talmon, “Common manifold learning using alternating-diffusion,” tech. rep., Yale, 2014.
- [3] R. R. Coifman and M. J. Hirn, “Diffusion maps for changing data,” *Applied and Computational Harmonic Analysis*, vol. 36, no. 1, pp. 79 – 107, 2014.
- [4] R. Talmon and R. R. Coifman, “Empirical intrinsic geometry for nonlinear modeling and time series filtering,” *Proc Natl Acad Sci USA*, vol. 110, pp. 12535–12540, Jul 2013.
- [5] A. Singer and H. Wu, “Vector diffusion maps and the connection laplacian,” *Communications on Pure and Applied Mathematics*, vol. 65, no. 8, pp. 1067–1144, 2012.
- [6] A. Schclar, A. Averbuch, N. Rabin, V. Zheludev, and K. Hochman, “A diffusion framework for detection of moving vehicles,” *Digital Signal Processing*, vol. 20, no. 1, pp. 111 – 122, 2010.
- [7] L. du Plessis, S. Damelin, and M. Sears, “Reducing the dimensionality of hyperspectral data using diffusion maps,” in *Geoscience and Remote Sensing Symposium IEEE International IGARS 2009*, vol. 4, pp. IV–885–IV–888, July 2009.
- [8] M. Belkin and P. Niyogi, “Laplacian eigenmaps for dimensionality reduction and data representation,” *Neural Comput.*, vol. 15, pp. 1373–1396, June 2003.
- [9] J. Lomask, *Seismic Volumetric Flattening and Segmentation*. PhD thesis, Stanford University, 2007.
- [10] D. Gao, “3d seismic volume visualization and interpretation: An integrated workflow with case studies,” *Geophysics*, vol. 74, no. 1, pp. W1–W12, 2009.
- [11] D. Parks, “Seismic image flattening as a linear inverse problem,” Master’s thesis, Colorado School of Mines, Golden, Colorado 80401, 2010.
- [12] R. R. Coifman and S. Lafon, “Diffusion maps,” *Applied and Computational Harmonic Analysis*, vol. 21, no. 1, pp. 5 – 30, 2006.
- [13] F. Woolfe, E. Liberty, V. Rokhlin, and M. Tygert, “A fast randomized algorithm for the approximation of matrices,” *Applied and Computational Harmonic Analysis*, vol. 25, no. 3, pp. 335 – 366, 2008.
- [14] N. Halko, P.-G. Martinsson, and J. A. Tropp, “Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions,” *ArXiv e-prints*, Sept. 2009.